

## Amendments to the Claims

1. (Currently Amended) A method for providing remote computer control of an application executing on a second computer from a first computer over a network, comprising:

~~via a first user interface of the first computer,~~ receiving a first user input instruction by a ~~proprietary first~~ operating system running on the first computer for execution, the first user input instruction being operationally compatible with a first computer language of the ~~proprietary first~~ operating system and operationally incompatible with a second computer language of a second operating system executing on the second computer ~~which incorporates a second user interface,~~ wherein the first user interface is dissimilar to the second user interface the first operating system being incompatible with the second operating system thereby requiring the first user input instruction to be translated from the first computer language of the first operating system in order to be executed by the second operating system;

at the first computer, translating the first user input instruction from the first computer language of the first operating system into a ~~non-proprietary~~ data script defining at least one XML item utilizing a first device driver resident in the ~~proprietary~~ operating system on the first computer, wherein the first device driver formats the first user input instruction into at least one XML item corresponding to the first user input instruction;

transmitting the ~~non-proprietary~~ data script defining the at least one XML item from the first computer to the second computer;

translating the ~~non-proprietary~~ data script defining the at least one XML item into a second user input instruction the second computer language of a second operating system utilizing a second device driver in the second operating system on the second computer, wherein the second device driver translates the at least one XML item corresponding to the first user input instruction into the second user input instruction, the second user input instruction being compatible with the second language of the second operating system running on the second computer and incompatible with the first computer language of the ~~proprietary first~~ operating system running on the first computer, the second user input instruction being functionally similar to the first user input instruction for execution on the second computer; and

executing the second user input instruction on the second computer.

2. (Previously Presented) The method of claim 1, wherein receiving the first user input instruction comprises receiving an instruction for outputting data.
3. (Previously Presented) The method of claim 2, wherein receiving the first user input instruction for outputting data comprises receiving an instruction for displaying data.
4. (Previously Presented) The method of claim 2, wherein receiving the first user instruction for outputting data comprises receiving an instruction for generating a sound.
5. (Previously Presented) The method of claim 1, wherein receiving the first user instruction comprises receiving an instruction for inputting data.
6. (Previously Presented) The method of claim 5, wherein receiving the first user instruction for inputting data comprises receiving an instruction indicating a mouse input.
7. (Previously Presented) The method of claim 5, wherein receiving the first user instruction for inputting data comprises receiving an instruction indicating a keyboard input.
8. (Previously Presented) The method of claim 1, wherein translating the first the first user instruction into a data script defining at least one XML item comprises generating a first XML tag defining the beginning of the XML item, generating a data item corresponding to the first user instruction, and generating a second XML tag defining the end of the XML item.
9. (Original) The method of claim 1, wherein transmitting the data to a second computer comprises transmitting the data using HTTP.
10. (Previously Presented) The method of claim 1, wherein translating the data into a second input instruction comprises identifying a first XML tag defining the beginning of an XML item, identifying a data item corresponding to a user input instruction, identifying a second XML tag defining the end of an XML item.

11. (Cancelled)

12. (Previously Presented) A computer readable medium having computer-implementable instructions stored thereon for performing the method recited in claim 1.

13-19. (Cancelled)

20. (Currently Amended) A system for remote computer access between computing systems with incompatible operating systems, comprising:

a first computing system having stored thereon software which when executed on the first computing system:

receives a user input via a first user interface of the first computing system;

identifies user input instructions generated by a ~~proprietary~~ operating system on the first computer system in a first computer language, the user input instructions relating to generating a system output via a second user interface of the first computing system in response to the user input,

translates the user input instructions into a first non-proprietary data script defining an outgoing software object corresponding to the user input instructions~~[[;]]~~, the translation being accomplished by a first device driver within the ~~proprietary~~ operating system on the first ~~computer~~ computing system,

transmits the outgoing software object to a second computing system, and

receives an incoming software object comprising a second non-proprietary data script from the second computing system reflecting a response to the user input instructions for execution on the second user interface of the first computing system, wherein the second ~~non-proprietary~~ data script is translated by the first device driver into a system output instruction in the first computer language being compatible with the ~~proprietary~~ operating system of the first computing system and being incompatible with a second computer language of a second computing system, the system output instruction then being executed on the first ~~computer~~ computing system as a system output via the second user interface.

21. (Currently Amended) A method for providing remote computer access, comprising:  
at a first computer, receiving outgoing instructions relating to generating an output on  
[[a]] the first computer from a first operating system in a first computer language executing on  
the first computer, the instructions being compatible with the first operating system and  
incompatible with a second operating system running on a second computer using a second  
computer language;

creating data defining a first XML item corresponding to the outgoing instructions  
~~relating to generating output on the first computer~~, wherein the outgoing instructions are  
translated into the first XML element at the first computer;

transmitting the first XML element from the first computer to the second computer;  
at the first computer, receiving data from the second computer defining a second XML  
item in response to the outgoing instructions ~~to generate output on the first computer from the~~  
~~second computer~~;

creating incoming instructions relating to generating the output on the first computer  
from the data defining the second XML item, wherein the incoming instructions are translated  
from the second XML item at the first computer and after which the incoming instructions are  
compatible with the first computer language of the first operating system running on the first  
computer and are incompatible with the second computer language of the second operating  
system running on the second computer; and

executing the incoming instructions to generate the output at the first computer.

22. (Previously Presented) The method of claim 21, wherein receiving incoming instructions  
relating to generating output comprises receiving instructions relating to generating visual or  
audio output.

23. (Previously Presented) The method of claim 21, wherein creating the first XML item  
corresponding to the outgoing instructions relating to generating output comprises generating at  
least a first XML tag defining the beginning of the first XML item, generating a data item  
corresponding to the instruction relating to generating output; and generating at least a second  
XML tag defining the ending of the first XML item.

24. (Previously Presented) The method of claim 21, wherein transmitting the data defining the first XML item comprises transmitting the data defining the first XML item using HTTP protocol.

25. (Previously Presented) The method of claim 21, wherein creating incoming instructions relating to generating the output comprises identifying a first XML tag identifying the beginning of the XML item, identifying a data item corresponding to an input, and identifying a second XML tag identifying the ending of the XML item.

26. (Currently Amended) A method for providing remote computer access between computing systems with incompatible operating systems, comprising:

receiving a first user input instruction relating to a user input received via a first user interface of the first computer by a first operating system on the first computer using a first computer language, the first user input instruction being compatible with the first operating system and incompatible with a ~~proprietary~~ second operating system on the second computer using a second computer language;

creating data defining a first software object in a non-proprietary format corresponding to the first user input instruction relating to the user input;

transmitting the first software object from the first computer to the second computer;

at the second computer, translating the first software object from the non-proprietary format to a second user input instruction compatible with the ~~proprietary~~ second operating system and incompatible with the first ~~proprietary~~ operating system;

executing the second user input instruction by the second computer;

receiving data from the second operating system related to the second user input instruction being executed, the data defining a first system output instruction, the first system output instruction relating to the first user input instruction and being compatible with the ~~first~~ second operating system executing on the second computer and incompatible with the first operating system on the first computer;

creating data defining a second software object in the non-proprietary format that corresponds to the second user input instruction;

transmitting the second software object from the second computer to the first computer;

at the first computer, translating the second software object to a second system output instruction being compatible with the ~~proprietary~~ first operating system and incompatible with the ~~proprietary~~ second operating system; and

executing the second system output instruction to render the user output by the first computer on a second user interface.

27. (Previously Presented) The method of claim 25, wherein transmitting the data defining the first and second software objects comprises using the HTTP protocol to transmit the first and second software objects.

28. (Previously Presented) The system of claim 20 wherein the first user interface is different from the second user interface.

29. (Previously Presented) The system of claim 26 wherein the first user interface is different from the second user interface.

30. (Currently Amended) The system for remote computer access between computing systems with incompatible operating systems of claim 20, further comprising the second computing system having stored thereon software which when executed on the second computing system:

- receives the outgoing software object from the first computing device;
- translates the first non-proprietary data script using a second device driver executing in conjunction with a second ~~proprietary~~ operating system executing on the second computer system into the user input instructions identified by the first computing system but operationally compatible with a second operating system executing on the second computer system and operationally incompatible with the ~~proprietary~~ operating system executing on the first computer system;
- executes the user input instructions compatible with the second operating system;
- identifies system output instructions operationally compatible with a second operating system executing on the second computer system and operationally incompatible with the ~~proprietary~~ operating system executing on the first computer

system, the system output instructions being responsive to the user input instructions identified by the first computing system,

translates the system output instructions into a second non-proprietary data script defining an incoming software object utilizing the second device driver

transmits the incoming software object; and

a communications network operably coupled between the first computing system and the second computing system for transmitting the first and second non-proprietary data scripts defining incoming and outgoing software objects between the first computing system and the second computing system.